217967US2

TITLE OF THE INVENTION

SETTING ERROR AVOIDABLE PRINTING SYSTEM AND METHOD

5

CROSS REFERENCE TO A RELATED APPLICATION

This application claims priority under 35 USC § 119 to
Japanese Patent Application No. 2001-011294 filed on
January 19, 2001, the entire contents of which being herein
10   incorporated by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to printing
15   systems that print a document or the like generated by a
terminal apparatus such as a personal computer connected to
a network such as a LAN, and more particularly, to printing
systems capable of preventing an operator from erroneously
setting printing conditions.

20

Discussion of the Background

A document printing system described, for example,
in Japanese Patent Application Laid Open No. 2000-298564
enables a printer driver to generate an intermediate file
25   described with a PDL (Page Description Language). In such
a system, application software sets and changes prescribed
print condition setting to the intermediate file so that
print data can receive optional processing. As a result,
highly sophisticated printing is enabled.

30   Further, a number of functions of a printer has
increased recently. For example, as described in Japanese
Patent Application Laid Open No. 11-143657, not only
magnification and integration printings, but also complex

1

printing, such as duplex printing, punching, stabling, etc., have been achieved only by prescribed application software and a printer driver function. However, depending upon a direction of a sheet to be printed, for example,

5   impermissible settings arise due to a physical limitation on the printer such that punching can not be performed at a designated position.

In general, before printing is performed using prescribed application software, a print condition setting

10   user interface is displayed by a printer driver using API (Application Program Interface) or the like provided by an OS (original system). Then, a combination of impermissible settings which are set through the print condition setting user interface is excluded from available options, for

15   example, so that erroneous print condition settings cannot be selected. Then, setting details are stored in a memory, and further sent to the printer driver in order to be printed.

Further, in Japanese Patent Application Laid Open

20   No. 2000-207150, a client apparatus generates a print job including data designating printing a document. A print server is also provided to interpret the print job and generate and provide printable data from the document to a printer, thereby enabling the printer to print. In such a

25   system, a user accesses a document supervisory database, included in the print server, from document supervisory application software which is included in the client apparatus. The user simultaneously makes a query if printable data that constructs the print job is stored. If

30   the printable data is stored, it is read and reused as print designation data.

According to the document printing system described

in Japanese Patent Application Laid Open No. 2000-298564, application software can recognize a type of function related to printing and settable from a printer driver, and settable details for the function. However, the inventor

5 of the present invention realized when a prescribed combination of setting details is generated as print condition settings, an inappropriate combination may be created depending upon contents of the combination. In general, only a printer driver which is specialized for a

10 printer can readily determine if such a combination is appropriate. Accordingly, when such a determination system is incorporated in application software, there is a disadvantage if the application software is not specialized for a printer.

15 Furthermore, a disadvantage may arise, for example, when "landscape" is set as print condition setting for a sheet in a printing apparatus as described in Japanese Patent Application Laid Open No. 11-143657. The inventor of the present invention realized that the disadvantage is

20 that an impermissible setting item, such as punching at an upper side of the sheet, disappears from the print condition setting user interface that preferably displays all the options for confirmation purpose.

In addition, according to the system of Japanese

25 Patent Application Laid Open No. 2000-207150, the printable data is taken in once by the client apparatus and then is transferred to the printer via a network again. However, the inventor of the present invention realized that a disadvantage arises because the load on the network

30 increases and therefore is time consuming. In addition, setting of a printer driver should be performed at each client apparatus.

3

SUMMARY OF THE INVENTION

Accordingly, an aspect of the present invention is to address and resolve the above and other problems and provide a new printing system. The above and other aspects are achieved according to the present invention by providing a novel printing system including a document supervisory client configured to generate one or more print condition settings, and a document supervisory server configured to perform printing. The document supervisory client may query the document supervisory server to determine if the print condition settings are appropriate. The document supervisory server may respond with regard to advisability of the print condition settings.

In another embodiment, if print condition settings are not appropriate, the document supervisory server may change a combination of print condition settings to provide appropriate print condition settings and send the appropriate print condition settings to the document supervisory client.

In yet another embodiment, a first computer including a document supervisory client may be provided so as to generate one or more print condition settings. A second computer including a document supervisory server may also be provided so as to perform printing. A printer may be connected to a network.

In yet another embodiment, the document supervisory server may send a setting value related to appropriate print condition setting together with character strings and graphs each representing the appropriate print condition setting to the document supervisory client as initial values. The document supervisory client may generate a

4

user interface for providing print condition settings based upon the setting values, character strings, and graphs.

In yet another embodiment, the document supervisory client may include a function of sending an ID which uniquely identifies a document stored in a database which is located in a document supervisory server. In addition, the document supervisory server may obtain an applicable document corresponding to the ID and execute printing.

BRIEF DESCRIPTION OF DRAWINGS

A more complete appreciation of the present invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Fig. 1 is a schematic configuration chart illustrating an exemplary embodiment according to the present invention;

Fig. 2 is a block chart illustrating a configuration of software of a document driver module;

Fig. 3 is a block chart illustrating a configuration of a printer driver module;

Fig. 4 is a schematic chart illustrating communication between a document supervisory client and a document supervisory server;

Fig. 5 is a schematic chart illustrating a display of a print dialog UI (User Interface);

Fig. 6 is a schematic chart illustrating a display of a print condition setting dialog UI;

Fig. 7 is a schematic chart illustrating a display of a print condition setting dialog UI that is changed;

5

Fig. 8 is a schematic chart illustrating a configuration of print data transmitted from the document supervisory server;

Fig. 9 is a chart illustrating a first table showing exemplary API functions and functions;

Fig. 10 is a chart illustrating a second table showing exemplary queries and returning values;

Fig. 11 is a chart illustrating a third table also showing exemplary queries and returning values;

Fig. 12 is a chart illustrating a fourth table showing exemplary queries and returning values;

Fig. 13 is a chart illustrating a fifth table showing exemplary functions and setting values; and

Fig. 14 is a chart illustrating a sixth table showing exemplary setting values and returned setting values.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout several views. In Fig. 1, a printing system may be provided and include a plurality of computers 2a, 2b, 2c and a plurality of printers 3a, 3b, 3c each connected to a network 1, for example, but not limited to a LAN. The computer 2a may include a document supervisory server 4 composed of document supervisory software, and a printer driver module band 5 that generates outputs to the printers 3a and 3b so that perform printing is performed. Each of the computers 2b and 2c may include a document supervisory client 6 including client software that generates print condition settings and accesses the document supervisory server 4 of the computers 2a.

As illustrated in Fig. 2, the document supervisory
server 4 may include a server I/F module 7, a document
supervisory control module 8, and a server print module 9,
in order to execute printing. The server I/F module 7 may
5   perform communication with each of the document supervisory
clients 6 of the computers 2b and 2c, illustrated in Fig.
1, via the network 1. Each of the document supervisory
clients 6 may perform command setting and data exchanging
with both of the document supervisory control module 8 and
10   the server print module 9 via the server I/F module 7. An
example in which one server I/F module 7 is provided will
be described herein below. However, the server I/F module
7 may be provided in each of the respective document
supervisory control module 8 and the server print module 9.
15   The document supervisory control module 8 may access, store
a new document, and read a document from the database 10.
The server print module 9 may access a printer driver of
the printer driver module band 5 and exchange data via an
interface of either the printer driver or a standard of an
20   OS. In addition, the server print module 9 may be
connected to the document supervisory control module 8 and
enabled to access indirectly the database 10.

Each printer driver module 11 of the printer driver
module band 5 may generate an intermediate file described
25   with a PDL. By giving and changing print condition
settings to the intermediate file, the printer driver
module 11 may enable print data to receive optional
processing in accordance with the print condition settings.
For example, as illustrated in Fig. 3, the printer driver
30   module 11 may include an intermediate PDL generation
processing section 12 that generates an intermediate PDL by
a conventional application software or other appropriated

7

means other than a printer driver. A PDL processing device 13 also may be provided so as to transmit the intermediate PDL and receive data to and from the server print module 9. Printer driver module 11 also may include a final PDL

5   generation processing section 14.

Fig. 4 illustrates exemplary communication performed in the printing system, constructed in the above-described manner, between the document supervisory client 6 of the computer 2b and the document supervisory server 4. Client

10 application software 61 may be provided in the document supervisory client 6, and may be initially connected to the server I/F module 7 via the network 1 and a route A. As a connection technology, a DCOM (Distributed Component Object Model) utilized in an OS such as, but not limited to,

15 Microsoft Windows® may be available. Subsequently, the client application software 61 may request the server print module 7 for a list of printer names capable of printing under control of the document supervisory control module 8 or server through route B. As one example, an interface of

20 the printer driver may be provided in the form of API.

The API may be provided as a DLL (Dynamic-Link Library). For example, the DLL file may be installed simultaneously when a printer driver is installed. The server print module 9 may attempt to load a DLL file.

25 When DLL file loading results in failure after attempting, a determination may be made that a printer corresponding to server printing does not exist, and an empty list of the printer names may be returned to the client application software 61. In contrast, when the DLL file is

30 successfully loaded, the entire names installed in the document supervisory server 4 for respective printers are obtained by the client application software 61. Then, the

8

client application software 61 may make a query if the server printing is corresponded using the API function "CheckServerPrint" listed in Table 1 of Fig. 9 in a manner as illustrated in the Table 2 of Fig. 10.

In response to the query by the client application software 61, the document supervisory server 4 may return a list including only names identifying printers capable of performing the server printing to the client application software 61. The client application software 61 may then display a print dialog user interface (UI) 15 including the list indicating the printer names as illustrated in Fig. 5. When a user confirms the list provided on the display, selects a prescribed printer, and depresses a print condition setting button 16, the client application software 61 may display a print condition setting dialog UI 17 as illustrated in Fig. 6. To obtain information displayable on the print condition setting dialog UI 17, a query about a kind of a print function settable in the printer may be made through routes "B" and "C". The query may be made, for example, with the API function "GetPrinterFunc" provided by the DLL of the printer driver illustrated in Table 1 of Fig. 9. Then, an API function, such as "PAPER/Sheet-Size", "LAYOUT/sheet-direction", "DUPLEX/Duplex", "NUP/Integration" "PUNCH/Punch", or "STAPLE/Staple" may be obtained as print function information.

Also, a query as to the kind of print condition setting values which may be modified or items settable for the print function already obtained may be made via the same routes. For example, as Fig. 12 illustrates in Table 4, a setting value, such as, but not limited to, "NONE", "LEFT", "RIGHT", "UP", etc., may be returned by the

9

application software 61 for the function "PUNCH" together with character strings and graphic data each representing the setting value on the UI.

Further, one or more setting values which may be modified for other functions may be obtained as illustrated in Table 5 of Fig. 13. In the table 5, both of the character strings and graphic data may be omitted.

By obtaining setting values which may be modified, an initial print condition setting value may be generated. A format of the initial print condition setting value may be generated in as a list, for example, PAPER=A4, LAYOUT=PORTRAIT, DUPLEX=NONE, NUP=NONE, PUNCH=NONE, STAPLE=NONE.

An inquiry to determine if optionally set or modified setting values constitute an appropriate combination may be made through the routes B and C of Fig. 4 with API function "SetCommand." When a determination that the optionally set setting values are not an appropriate combination, a correction may be made to provide an appropriate combination as illustrated in Table 6 of Fig. 14. For example, a printer driver may determine the appropriate combination via a table representing appropriate combinations. Also, any of the setting values may be interrelated such that they may effect the ability to modify other setting values. For example, a setting value "UP" set for the function "PUNCH," affect the setting values for the function "PAPER" and the setting value "LANDSCAPE" for the function "LAYOUT." For example, if "UP" is set then "LANDSCAPE" should be set for the function "LAYOUT". However, if the setting value "PORTRAIT" is set for the function "LAYOUT", a determination or advisability may be made that such print condition settings are

10

inappropriate.

A change may be then executed on at least one setting value. For example, simultaneously, an operation such as changing "UP" to "LEFT" may be performed for the "PUNCH" by referring to Table 6 of Fig. 14, therefore providing appropriate setting values.

A determination or advisability of which setting value to change, may be based on a priority order of the function. Information related to the priority order may be previously stored with the above-described table 6 of Fig. 14 representing the appropriate setting combination. For example, information related to a method for determining the priority and the appropriate setting combinations may be included in a printer driver. Such information may be included when the printer driver is designed.

Further more, determining that one or more setting values which may be changeable from currently set print condition setting values and that an appropriate selection is maintained, may be queried through the routes B and C of Fig. 4, for example, using the API function "GetFuncValue". In response to the query, since the print condition setting value has already been set when the API function "SetCommand" is sent, at least one appropriate setting value may be returned in response thereto. For example, the appropriate setting values of "NONE", "LEFT", and "RIGHT" may be returned for the function "PUNCH", and therefore provided as setting values which may be selected. In other words, the previously returned "UP" may not be returned. In this way, one or more setting values which may be modified for a function may be obtained. Then, based upon obtained information, the print condition setting dialog UI 17 of Fig. 6 may be displayed. In the

print condition setting dialog UI 17, the appropriate print condition settings previously selected may be maintained as initial setting values as shown by a black circle in Fig. 6. In addition, setting values not appropriate either may

5   be erased with double lines or may be entirely erased so that a user can not select.

As illustrated in Fig. 7, when a user has confirmed the list provided on the display and changes in the setting values, the display of the print condition setting dialog

10   UI 17 may be updated and changeable setting values for the functions may be obtained again. For example, by selecting "landscape" for "Sheet Direction", the user may be then allowed to change a setting value such as "punch", "Staple", etc. By changing print condition setting in this

15   way, and activating an "OK" button 18 on the print condition setting dialog UI 17, the print condition settings may be confined and reset.

When print execution is designated after the "OK" button 19 is activated, the client application software 6

20   may transfer both a document ID for identifying a document which is designated for printing and a print condition setting that is set to the server print module 9 through route D of Fig. 4. The server print module 9 may obtain document data from the database 10 through routes "E" and

25   "F" while indicating the document ID. The document data thus obtained may be transmitted to the printer driver module band 5 through a route G. The printer driver module band 5 may then process the document data and generate an intermediate PDL. The printer driver module band 5 may

30   send the intermediate PDL to the server print module 9. Multiple intermediate PDLs may be generated when there exists a plurality of printing objective data. The server

12

print module 9 may generate and send again the data 20 illustrated in Fig. 8 by merging the print condition setting with the intermediate PDL to the printer driver module band 5.   The printer driver module band 5 may

5   generate a final PDL based upon the data and indicate printing by carrying the data to a print spooler (not shown).

The mechanisms and processes set forth in the present invention may be implemented using one or more

10   conventional general purpose microprocessors and/or signal processors programmed according to the teachings in the present specification and is appreciated by those skilled in the relevant arts. Appropriate software coding may be prepared readily by skilled programmers based on the

15   teachings of the present disclosure and is apparent to those skilled in the relevant arts. However, as readily apparent to those skilled in the art, the present invention also may be implemented by the preparation of application-specific integrated circuits by interconnecting an

20   appropriate network of conventional component circuits or by a combination thereof with one or more conventional general purpose microprocessors and/or signal processors programmed accordingly. The present invention may also include a computer-based product which may be hosted on a

25   storage medium and includes, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnet-optical disks, ROMs, RAMs, EPROMs, EEPROMs, flash memory, magnetic or optical cards, or any type of media suitable for storing electronic instructions.

30   Numerous additional modifications and variations of the present invention are possible in light of the above teachings.   It is therefore to be understood that within

13

the scope of the appended claims, the present invention may
be practiced otherwise than as specifically described
herein.